**Part 2**

1. In Errors 4, there was about 5 errors. In Errors 5, there was about 4 errors. In Errors 6, there was about 11 errors.
2. My testing strategy of the modified program after it runs was to check if each of the programs worked and to be careful to see if each of them would follow what they needed to do. Using the right operators come in handy.

**Part 4**

* 4 + 5 \* 6 + 7 / 8
  + In order to find the result for this expression is we have to follow PEMDAS. Since there is no parenthesis, we don’t need to worry about them. So, the first thing you would need to do is do the multiplication between 5 times 6 and then divide 7 by 8. Afterwards, we would add starting from the left between 4 and the product of 5 times 6. Finally, we would add the sum of the last step with what 7 divided by 8 is and get we the final result.
* 4 + 5 \* 6 < 7 / 8
  + Similar to the last expression, we would start off with the multiplication between 5 and 6 and then move on to the division between 7 and 8. Afterwards, we would add 4 and the product of 5 times 6. Finally, we would use the < between the sum and the quotient to get the result.
* 4 + 5 > 6 && 7 < 8
  + There are no parenthesis or multiplication/division, we would have to check if there is any addition or subtraction. Since there is only addition on the left side, we would find the sum of 4 and 5. Afterwards, we would have to do the sum of 4 and 5 > 6 and then 7 < 8, left to right. And lastly, we would do && starting from the left to the right.
* 1 && 0
  + Since there is only one operation to do, we would evaluate the expression by doing the &&. And the result would undefined.